IN THE CLAIMS

1	1. (Original) A method for managing a code sequence, comprising:
2	processing a first set of sample values with coefficients from a first set of code sequence
3	coefficients to determine first partial accumulation results during a first time step;
4	processing a second set of sample values with coefficients from a second set of code
5	sequence coefficients to determine second partial accumulation results during a second time step;
6	processing the second set of sample values with coefficients from the first and second set
7	of code sequence coefficients to determine third partial accumulation results during the second
8	time step;
9	generating a lag result for a first sequence of sample values in response to the first and
10	second partial accumulation results; and
11	generating a lag result for a second sequence of sample values in response to the first and
12	third partial accumulation results.
1	2. (Currently Amended) The method of Claim 1, further comprising:
2	processing a third set of sample values with coefficients from a third set of code
3	sequence coefficients to determine fourth partial accumulation results during a fourththird time
4	step; and
5	updating the lag result for the second sequence of sample values in response to the first,
6	third, and fourth partial accumulation results.
1	3. (Original) The method of Claim 1, further comprising determining a synchronization
2	point for the code sequence from the lag results for the first and second sequence of sample
3	values.

- 4. (Currently Amended) The method of Claim 3, wherein determining a synchronization point comprises determining a lag result from the first and second sequences of sample values having the highest numerical value.
- 5. (Original) The method of Claim 1, wherein the first and second set of code sequence
 coefficients are contiguous coefficients from the code sequence.
- 6. (Original) The method of Claim 1, wherein the first and second set of sample values
 are contiguous sample values in a received sample.
 - 7. (Currently Amended) The method of Claim 1, wherein generating to determine first partial accumulation results from athe first set of sample values and coefficients from athe first set of code sequence coefficients during athe first time step comprises taking the products of the first set of sample values and the coefficients from the first set of code sequence coefficients.
 - 8. (Original) The method of Claim 1, wherein generating the lag result for the first sequence of sample values in response to the first and second partial accumulation results comprises taking a sum of the first and second partial accumulation results.
- 9. (Currently Amended) A method for managing a code sequence, comprising:
 accessing a first set of n coefficients in the code sequence and a first set of n sample
 values in a <u>first sample sequence</u> during a first time step;
- processing the first set of n sample values with coefficients in the first set of n
 coefficients to determine first partial accumulation results;
- accessing a second set of n coefficients in the code sequence and a second set of n
 sample values in the <u>first sample sequence</u> during a second time step;

1

2

3

4

1

2

3

8	processing the second set of n sample values with coefficients in the second set of n
9	coefficients to determine second partial accumulation results; and
10	generating a lag result for athe first sample sequence from the first and second partial
11	accumulation results.
1	10. (Original) The method of Claim 9, further comprising:
2	processing the second set of n sample values with coefficients in the first and second set
3	of n coefficients to determine third partial accumulation results; and
4	generating a lag result for a second sample sequence from the first and third partial
5	accumulation results.
1	11. (Original) The method of Claim 10, further comprising:
2	accessing a third set of n sample values in the sample during a third time step;
3	processing the third set of n sample values with coefficients in the second set of n
4	coefficients to determine fourth partial accumulation results; and
5	updating the lag result for the second sample sequence with the fourth partial
6	accumulation results.
1	12. (Original) The method of Claim 9, wherein the first and second set of n coefficients
2	are contiguous code sequence values in the code sequence.
1	13. (Original) The method of Claim 9, wherein the first and second set of n sample
2	values are contiguous sample values in the sample.
1	14. (Original) The method of Claim 9, wherein processing the first set of n sample
2	values with coefficients in the first set of n coefficients to determine the first partial

- 3 accumulation results comprises taking the products of the first set of n sample values and the 4 coefficients in the first set of n coefficients. 1 15. (Currently Amended) The method of Claim 9, wherein determining generating the 2 lag result for the first sample sequence from the first and second partial accumulation results 3 comprises taking a sum of the first and second partial accumulation results. 1 16. (Currently Amended) A method for managing a code sequence, comprising: 2 accessing sets of n contiguous sample values that include sample values in a plurality of 3 sample sequences; 4 accessing sets of n contiguous coefficients; and 5 processing in parallel the sample values in each of the plurality of sets of sample values 6 that are accessed in parallel with corresponding coefficients that are accessed, where each of the 7 plurality of sets of sample values are processed during a different time step. 1 17. (Original) The method of Claim 16 further comprising generating lag results for 2 each of the sample sequences. 1 18. (Original) The method of Claim 16, wherein each of the sets of n contiguous sample
- 1 19. (Original) The method of Claim 16, wherein each of the sets of n contiguous coefficients is accessed at a different time step.

values is accessed at a different time step.

2

- 20. (Original) The method of Claim 16, wherein processing the sample values in each of the plurality of sets of sample values with corresponding coefficients comprises generating partial accumulation results.
- 1 21. (Currently Amended) A correlator unit, comprising:
- a plurality of n sample sequence registers that store sample values from a plurality of
 sample sequences that are processed in parallel, the plurality of n sample sequence registers
 storing sample values from one set of sample values of a plurality of sets of sample values from
 the plurality of sample sequences at a time;
- a plurality of 2n code sequence registers that store up to 2n coefficients from a code
 sequence; and
- a processing unit that processes the sample values in each of the plurality of sets of
 sample values in the plurality of n sample sequence registers in parallel with corresponding
 coefficients in the plurality of 2n code sequence registers, where each of the plurality of sets of
 sample values is processed during a different time step.
- 22. (Original) The correlator unit of Claim 21, wherein the processing unit comprises
 an addition-multiplication tree.
- 1 23. (Original) The correlator unit of Claim 22, wherein the addition-mulitplication tree comprises:
- 3 a plurality of specialized multiplexers; and
- 4 a plurality of adders.
- 24. (Original) The correlator unit of Claim 23, wherein each of the specialized
 multiplexers, comprises:

6

- 3 a multiplexer; and
- 4 a plurality of circuits that perform an XOR function.